REMARKS

Applicant has studied the Office Action dated June 24, 2005. It is submitted that the application is in condition for allowance. Claims 1-23 are pending. Reconsideration and allowance of the pending claims in view of the following remarks is respectfully requested.

In the Office Action, the Examiner rejected claims 1-23 under 35 U.S.C. § 102(e) as being anticipated by Hobbs (U.S. Patent No. 6,523,022). The claims have not been amended to overcome the cited prior art.

Before discussing the prior art in detail, it is believed that a brief review of the invention as claimed, would be helpful. Independent claims 1 and 11 recite, *inter alia*:

> • associating an associative array with one or more entries to each of a plurality of UI objects presentable as part of a UI;
>
> presenting at least one UI object based on a state of at least one global context flag for the UI;
>
> receiving at least one of a response from an end-user to the presentation of the at least one UI object and an event-based trigger;
>
> altering the state of the global context flag label based on the response from the end-user;
>
> performing a Boolean comparison between the global context flag and one or more of the entries in the associative array for each of the UI objects; and
>
> presenting zero or more of the plurality of UI objects as part of the UI to the end-user based upon a result of the Boolean comparison.

The present invention discloses a method, system, and computer readable medium for programming a UI (User Interface) based on global context flags. The global context flags are organized in an associative array. The particular state of the program dictates the values within the global context flag array. These flags in turn are used in a Boolean operation to select which UI objects should be rendered and/or made available to the end-user. Programming the desired UI objects for usage is accomplished by attaching attributes for the desired rendering which, if favorably compared using a Boolean operation to the global context flags, will result in their selection or deselection.

In the present invention, at least one menu list 318-328, 330-336, 338-344, etc. is associated with an application, for example, Microsoft™ Word™. See FIG. 3 and page 11, line 18 through page 13, line 2. The menu list includes one or more choices, such as "File", "New", "Open", "Close", etc. See FIG. 3. Importantly, each choice is associated with an associative array (306-310 in FIG. 3). The choices available to a user vary dependent upon the Boolean matching of the associative array entries to the global context flags assigned to each document.

The term "associative array" is a well-known term of art in the computer software field. An "associative array" is defined on page 11 of the specification in the present application as "a set of items, which are randomly accessible by a key, often a string." The global context flags 304 are nested in structure and usage as shown in FIG. 3 and described on page 11, lines 21-22. Therefore, the global context flag is also an associative array and the present invention is Boolean matching <u>two</u> associative arrays.

As an example, take the situation where a computer is used to open a version of Microsoft™ Word™. Each document opened in Word™ has with it properties which describe the state of the document. These properties are indicated by "global context flags," such as the document state (open or closed), the document type, and the presence of objects within the document. For instance, if just the Word™ application is running, and no document is open, (See FIG. 3) the global context flags will be document=closed, document type=none, and document object=none. These are presented to three possible menu lists, in this example, file 318, table 330, and graphic 338. Each list has an associative array attached, which defines when the object should be shown. By comparing the global context flags to the associative array using a Boolean AND function, the menu list items that should be shown to a user are determined. For instance, if a document isn't opened, it wouldn't make sense to present a user with an option to "Save."

The present invention provides a large advantage over the prior art due to the fact that programmers no longer have to assign all attributes to each state of a program during programming. The present invention allows the state of the program to dynamically determine the available options.

The cited Hobbs reference is dissimilar to the present invention. Hobbs discloses a method of dynamically augmenting the contents of a file in a database stored on a network resource. It is submitted that the subject matter of Hobbs is not even related to that of the present invention.

Specifically, the Hobbs reference is an information retrieval tool that utilizes a client device coupled to three servers: a Document Server, an Application Server, and a Database Server. Hobbs, col. 11, lines 34-49 and col. 15, lines 30-33. The three servers are interconnected over a network. Hobbs, col. 11, lines 49-53. The client device supports a browser that establishes connection to the servers. Hobbs, col. 15, lines 33-35.

In Hobbs, a user, employing the browser on the client, clicks on (or otherwise selects) a hypertext link on a page in order to send, via an HTTP network protocol, a request for an HTML file to the Document Server. Hobbs, col. 17, lines 19-22. An example HTTP statement is in the form http://www.example.com/file.html. In addition to the HTTP statement, an argument symbol is added to the end of the statement after a question mark (?). Hobbs, col. 17, lines 22-42. The HTTP network address and the argument symbol, e.g. "AR1", constitute the request for the linked term. Hobbs, col. 17, lines 42-44.

The browser then passes the request, including the argument symbol, to the Application Server. Hobbs, col. 27, lines 59-61. The argument symbol is ultimately used find a link to an address of a database where the requested information resides. The disclosed methods of locating the address are matching the argument to values in a look-up table, a hash table, an associative array, or a linked list. Hobbs, col. 11, lines 56-58. In the case of the associative array, the argument is a used as a key. Hobbs, col. 18, lines 4-6.

Alternatively, the argument that acts as a key or the key can be used to create a pop-up or floating window display for viewing by the user. Hobbs, col. 18, lines 22-24. The window display presents the user with a menu of choices for further areas of research pertaining to the key and, therefore, the linked term or terms. Hobbs, col. 18, lines 24-26. Each choice corresponds to an argument symbol. The user chooses one of the options presented, which will send an argument symbol to the Application Server. The argument symbol is used as a key in a table lookup on the Application Server. The table lookup then matches the key with one of a plurality of expert-predetermined optimum values used to retrieve records from a Data Warehouse or database. Each expert predetermined optimum value includes a network address for the Database Server, a query argument, and an authentication argument. Hobbs, col. 18, lines 22-30.

Therefore, Hobbs does not "present[...] at least one UI object based on a state of at least one global context flag for the UI," as recited in independent claims 1 and 11 of the instant application. The Examiner on page 2 of the Office Action compares this step to "argument acts as a key." While it is true that in Hobbs, the argument does act like a key, the arguments do not have states. The argument is simply a static link to an HTTP network address. As a result, it is not possible for UI objects to be "based on a state" of the arguments.

In addition, Hobbs does not disclose "altering the state of the global context flag label based on the response from the end-user," as recited in independent claims 1 and 11 of the instant application. In fact, Hobbs is completely silent on altering the state of anything. In Hobbs, a user selects a hyperlink and, in one embodiment, makes a second choice of further areas of research pertaining to the hyperlink chosen. Hobbs, col. 18, lines 22-30. No states are changed.

Finally, Hobbs does not disclose "presenting zero or more of the plurality of UI objects as part of the UI to the end-user based upon a result of the Boolean comparison," as recited in independent claims 1 and 11 of the instant application. Instead, Hobbs discloses selecting and linking to multimedia information, such as video, audio, graphics and text residing on a plurality of Data Warehouses through the internet or other network. Hobbs, abstract. The objects linked to in Hobbs are independent of any UI i.e. independent multimedia files not related to the state of the UI, and are thus, not "part of the UI."

Independent claims 8 and 18 recite limitations similar to claims 1 and 11. The Examiner cites 35 U.S.C. § 102(e) and a proper rejection requires that a single reference teach (i.e., identically describe) each and every element of the rejected claims as being anticipated by Hobbs.[1] Because the elements in independent claims 1, 8, 11, and 18 of the instant application are not taught or disclosed by Hobbs, the apparatus of Hobbs does not anticipate the present invention. The dependent claims are believed to be patentable as well because they are all ultimately dependent on either claim 1, 8, 11, or 18. Accordingly, the present invention distinguishes over Hobbs for at least this reason. The Applicant respectfully submits that the Examiner's rejection under 35 U.S.C. § 102(b) has been overcome.

CONCLUSION

The remaining cited reference (Sherlock et al.) has been reviewed and is not believed to affect the patentability of the claims as amended.

Applicant acknowledges the continuing duty of candor and good faith to disclosure of information known to be material to the examination of this application. In accordance with 37 CFR §1.56, all such information is dutifully made of record. The foreseeable equivalents of any territory surrendered by amendment are limited to the territory taught by the information of record. No other territory afforded by the doctrine of equivalents is knowingly surrendered and everything else is unforeseeable at the time of this amendment by the Applicant and his attorneys.

[1] See MPEP §2131 (Emphasis Added) "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). "The

Applicant respectfully submits that all of the grounds for rejection stated in the Examiner's Office Action have been overcome, and that all claims in the application are allowable. No new matter has been added. It is believed that the application is now in condition for allowance, which allowance is respectfully requested.

PLEASE CALL the undersigned if that would expedite the prosecution of this application.

Respectfully submitted,

Date: August 4, 2005

By: ______________________
Jon Gibbons, Reg. No. 37,333
Attorney for Applicant

FLEIT, KAIN, GIBBONS, GUTMAN BONGINI & BIANCO P.L.
551 N.W. 77th Street, Suite 111
Boca Raton, FL 33487
Tel (561) 989-9811
Fax (561) 989-9812

Please Direct All Future Correspondence to Customer Number **23334**

identical invention must be shown in as complete detail as is contained in the ... claim."